EXHIBIT C

PART 2

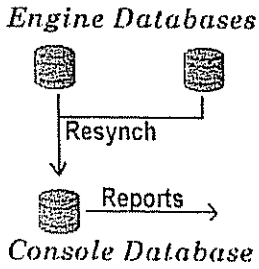


Figure 21. "Synchronize All Logs"

Synchronize All Logs

All information being logged is performed by the local database. Reports, however, are generated from the console database, not the Engines. To get information from the Engine to the Console, you must perform "Synchronize All Logs".

Console-Engine Communication

Data sent from the Management Console to the Engines includes the following:

- Start, stop and pause commands via the Engine menu.
- Changes to filter rules, attack signatures, and event responses.
- Keep-alive checks, shown as "Pings" in the Status column of the Engines window.
- Performance information, shown in the Traffic/(Pkts per second) column in the Engines window.

The RealSecure Management Console

The RealSecure for Windows NT Management Console displays the network data in a variety of ways, to help assess the network traffic and plan responses to events. This section describes the Console in detail.

Using the Management Console

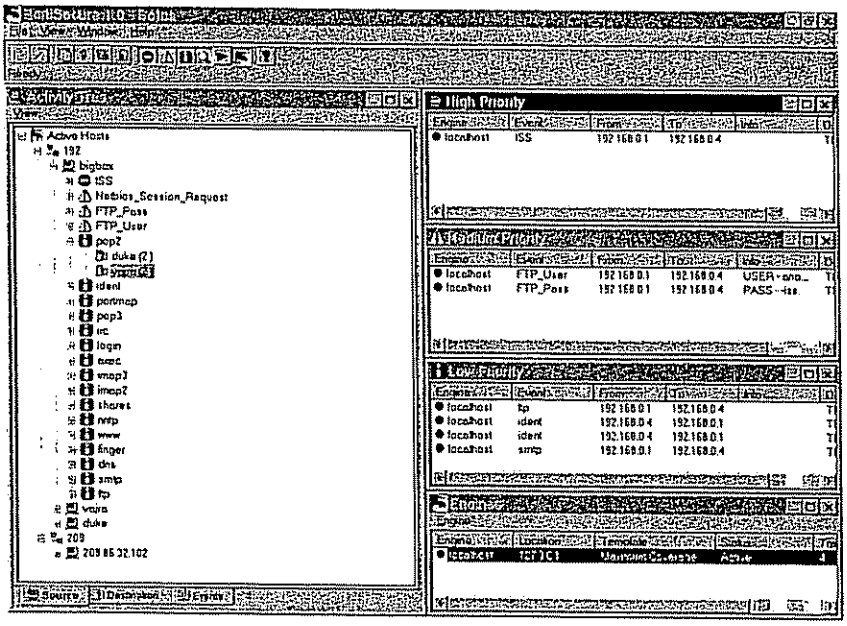


Figure 22: Management Console

The entire RealSecure Console is displayed. The following sections describe the components of the Console.

RealSecure Menus

The following section describes RealSecure's menus, with a brief description of each option.

File Menu

The following options are available from the File menu:

Import RS UNIX Config Files: This option creates a template using previously configured templates from RealSecure for UNIX. It displays the "Build Template from Files" dialog box.

Maintain Log: This option allows you to manage the Console database. It displays the "Maintain Console Log dialog box".

Synchronize All Logs: Causes the databases for all managed Engines to be uploaded to and integrate into the Console database.

Print Setup: Allows for the configuration of your printer.

Recent File: Displays a list of recently saved configurations.

Exit: Closes RealSecure for Windows NT.

View Menu

The following options are available on the Console's View Menu:

Toolbar: Toggles the Toolbar.

Status Bar: Toggles the Status Bar.

Activity Tree: Toggles the Activity Tree window.

Active Engines: Toggles the Engines window.

High Priority: Toggles the High Priority window.

Medium Priority: Toggles the Medium Priority window.

Low Priority: Toggles the Low Priority window.

Session Playback: Toggles the "Session Playback" window.

Display Key: Displays the "ISS Key Contents" dialog box.

Reports: Displays the "Reports" dialog box.

Options: Displays the "Console Configuration" dialog box.

Refresh: Updates the display in the Activity Tree.

Window Menu

The following options are available on the Window Menu:

Arrange Windows: Returns RealSecure's window layout to default settings

Note: *The RealSecure Console is designed for displays with a minimum resolution of 800x600 pixels.*

Help Menu

The following options are available from the Help Menu:

Help Topics: Displays the Contents page for the On-line Help System.

About RSNT: Displays version and copyright information about RealSecure for Windows NT

Engine Menu

At least one Engine must be added to the RealSecure Management Console before the Engines menu options become fully available. These options can also be accessed by selecting and right-clicking on an Engine from within the list of Active Engines.

The following options are available from the Engines menu in the Engines window:

New: Adds a new Engine which reports network Attack and Event data to the Management Console. Displays the "Add Engine" dialog box.

Start: Starts the currently selected Engine.

Ping: Checks to see if the currently selected Engine is Active.

Pause: Pauses the currently selected Engine.

Resume: Resumes activity on the currently paused Engine.

Shutdown: Stops the currently selected Engine.

Remove: Removes the currently selected Engine from the list of available Engines.

Note: *Removing an Engine only removes it from the list of managed Engines. The Engine will continue to run until it is stopped from a Management Console or at the monitor of the host on which the Engine is running.*

Maintain Log: Allows you to manage the Engine database. Displays the "Maintain Engine" dialog box.

Properties: Allows you to manage Engine configuration. Displays the Engine Properties dialog box.

Activity Tree View/Session Playback Menu

The following options are available from the View menus in the Activity Tree view from within the Session Playback window:

Show Names: When this option is selected, events in this window are viewed by machine name rather than by IP Address.

Inspect Event: Displays the "Event Inspector" window. This allows you to get more information about a specific network Event.

Expand All: Expands the view to show all levels of activity.

Collapse All: Collapses the view completely.

The RealSecure Toolbar

The Console's Toolbar contains the following options, as they appear from left to right:

Add Engine: When the Engines window is active, this button opens the "Add Engine" dialog box. The "Add Engine" toolbar button is grayed out when the Engines windows are inactive

View Engine Properties: Opens the "Engine Properties" dialog box. This button will be active only if an Engine in the Engines window is selected

Configuration Options: Opens the Console Configuration dialog box.

Display Key: Displays the current key.

Synchronize All Engine Logs With Console Log: Transfers all records from all Engine databases to the Console log database. The records are deleted from the Engine logs as they are transferred

View Reports: Opens the "Reports" dialog box.

High Priority View: Shows or hides the "High Priority" window.

Medium Priority View: Shows or hides the "Medium Priority" window.

Low Priority View: Shows or hides the "Low Priority" window.

Activity View: Shows or hides the "Activity Tree" window.

View Session Playback Window: Opens the "Session Playback" dialog box.

Engine View: Shows or hides the Engines window.

About: Displays copyright information for RealSecure.

Note: *The High, Medium and Low Priority windows contain the same information as the Activity View (Activity Tree window), but they are only time stamped and displayed serially.*

Main Banner Window

Located at the top of the RealSecure Main Window is the Main Banner Window. This area contains the menu bar, configuration buttons.

ISS Key Contents

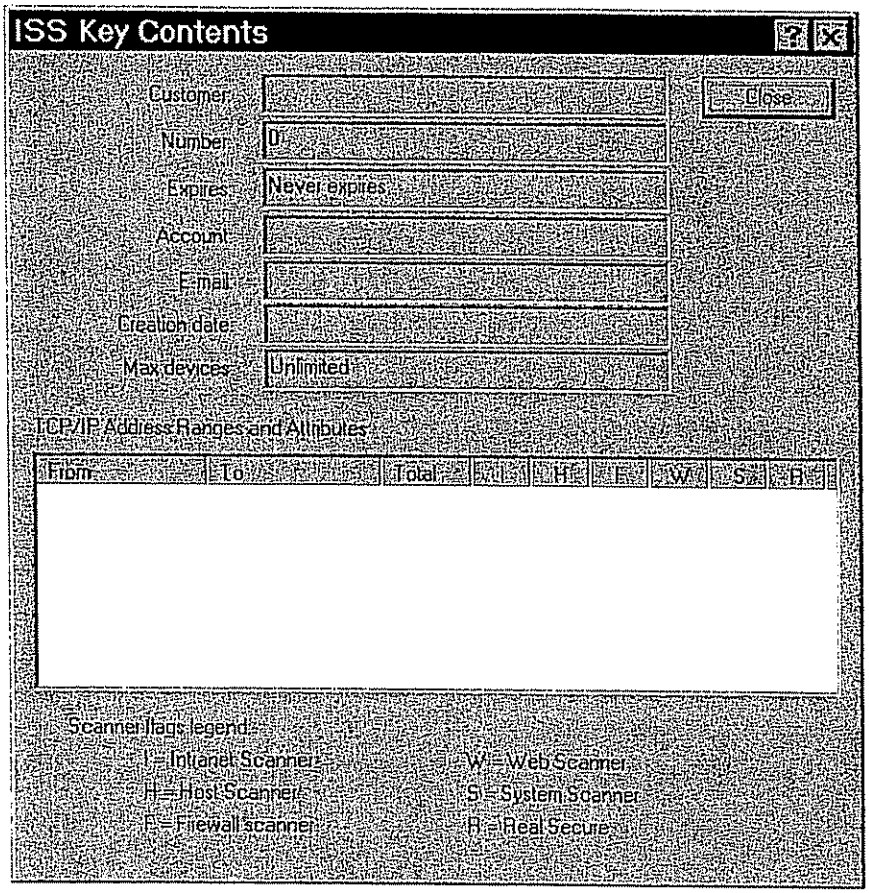


Figure 23: ISS Key Contents screen

To access the ISS Key Contents screen simply click on the icon that depicts a key from within the Toolbar. The ISS Key Contents screen displays the contents of the currently loaded key. RealSecure has built-in protection to ensure that it can only operate on intended networks which are licensed. The Windows NT version searches for keys in the same directory the program executables are installed. The default is `iss.key`, and is located in the same directory as the Management Console (`rsnt.exe`).

Customer: The name of the customer for whom the key was created.

Number: The ISS customer number.

Expires: Expiration date of the key. The software will not operate after this date with the current key. To obtain a new key, call ISS at 800-766-2362 or send e-mail to `sales@iss.net`.

Account: The ISS account name.

E-mail: The e-mail address.

Creation Date: Date and time the key was created.

Maximum Devices: Number of Engines that can be used.

TCP/IP Address Ranges: RealSecure Engines are restricted to the ranges which are enabled in the key. The enabled features are described by the following flags:

- **I** = Internet Scanner
- **H** = Host Scanner
- **F** = Firewall Scanner
- **W** = Web Scanner
- **S** = System Scanner
- **R** = Real Secure

Activity Tree Window

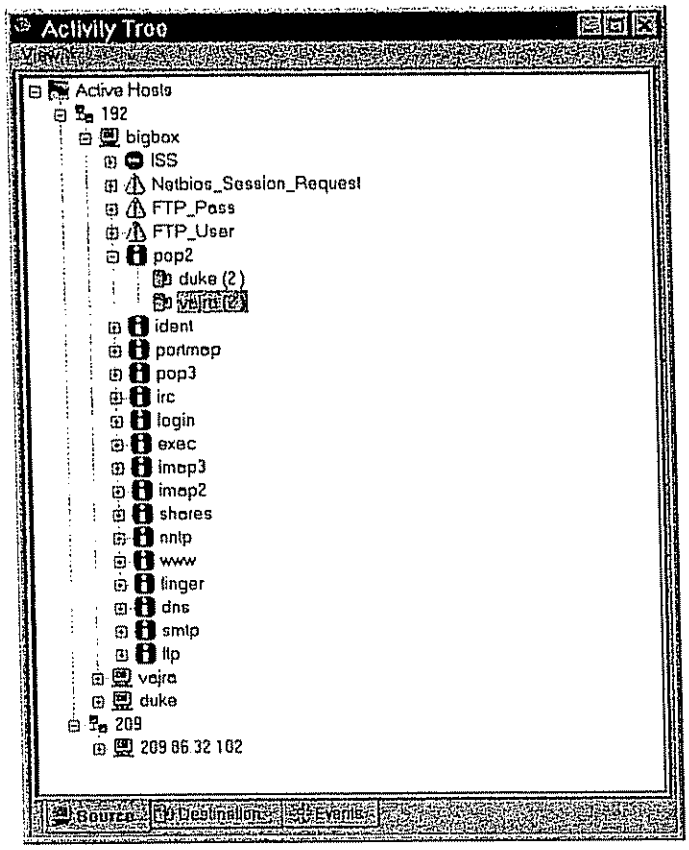


Figure 24: Activity Window – Source tab

This window, shown with the "Source" tab selected, displays the most recent network activity sorted by the addresses of the systems that initiated the activity. If the Show Names option is selected in the "Activity Window's View" menu, then the machine name associated with that IP address is displayed in the "Source Tree" window, rather than the numeric IP address.

In each case, you can click on the plus sign to the left of the entry to drill down for more information. You will generally see information associated with the rest of the session, including the source, destination, event type, and any information associated with the event. Right click on an event to Inspect the Event.

At the top of the activity screen, the View menu option allows you to toggle between numeric IP addresses and resolved domain names.

Two other tabs are available from the "Activity Tree" window:

Destination Tab

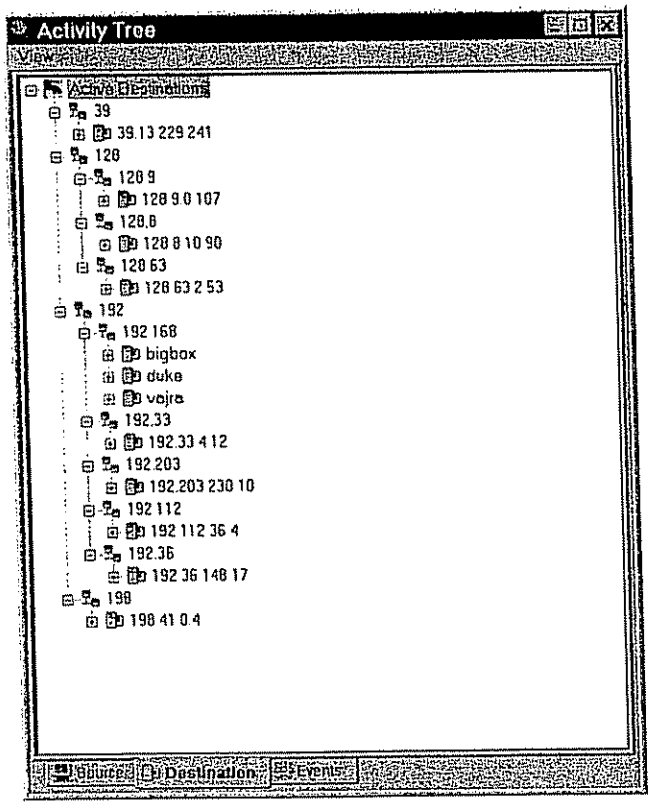


Figure 25: Activity Window – Destination tab

The "Destination" tab displays the most recent network activity sorted by the addresses of the systems that were the target of the activity.

Events Tab

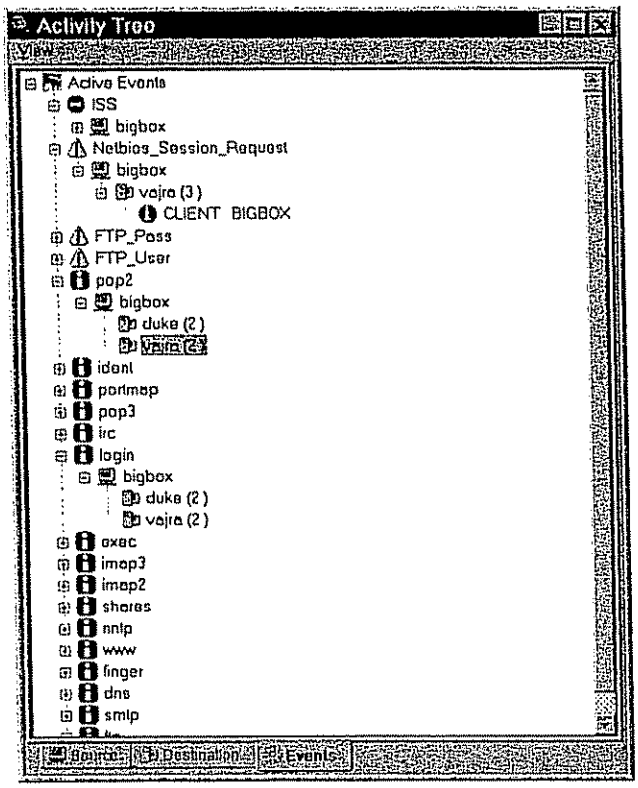


Figure 26: Activity Window – Events tab

The "Events" tab displays the most recent network activity sorted by the type and priority of the event as recognized by the RealSecure Engines. Events are sorted by High, Medium, or Low priority.

Note: *To receive more information, right click on the leaf of the tree. Information will vary for each Event and may not always be available for every Event. Whether or not information is available depends on the specific type of network Event. Information is indicated by the international information symbol.*

Information is indicated for the Events tab as follows:

Event Tab:
Source
Destination
Info (if available)

Information is indicated for the Source tab as follows:

Source Tab:
Source
Event
Destination
Info (if available)

Information is indicated for the Destination tab as follows:

Destination tab:
Destination
Event
Source
Info (if available)

RealSecure recognizes two types of network occurrences:

1. **Attacks** are network activity patterns indicating that someone may be engaged in unauthorized or undesirable activity involving the systems and/or data on your network. Examples of these include SATAN scans, ping floods, WinNuke, SYN floods, IP half scans, and attempts to obtain unauthorized root access.
2. **Decodes** are non-attack network activity that may be of interest to the Security Administrator. Examples of these include HTTP activity (i.e., who is surfing the Internet and where they are going?), analysis of access to Windows shares (e.g., connections from engineering to accounting), and e-mail session decoding.

RealSecure is shipped with the most comprehensive set of attack recognition patterns in the industry.

See also: Event Inspector
Configuring Engines
Templates

Engines Window

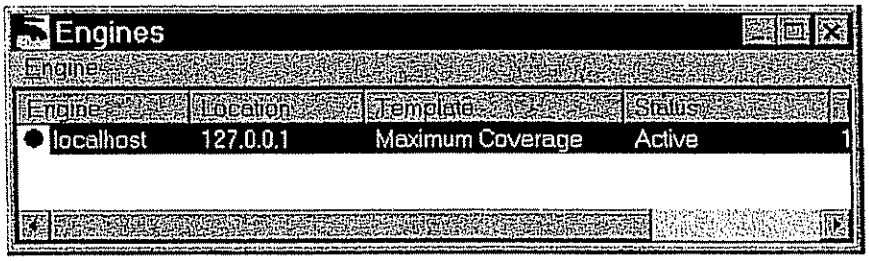


Figure 27: Engines Window

The Engines window displays information about the Engines that are currently running on RealSecure. The information is displayed in the following columns:

Engine: The name of each Engine that is currently being managed by the Management Console.

Location: The IP Address of each Engine that is currently being managed by the Management Console.

Template: The name of the Template each active Engine is currently using.

Status: The communications status of each active Engine.

Traffic (Pkt/Sec): The amount of network traffic, in packets per second, currently being handled by each Engine.

Priority Views

The High, Medium and Low Priority windows show the events and attacks that are reported by the current Engine.

Column Headers

The column headers in the "Priority" windows display data according to the filter rules established in the Templates. To refresh the events showing in the "High Priority" window, click on a column header.

Engine: The machine name of the reporting Engine

Event: The type of event being reported.

From: The IP address of the source machine associated with the network Event.

To: The IP address of the destination machine associated with the network Event.

Info: Displays information associated with the event, if applicable.

Date: The time and date of the event's occurrence.

Note: *By clicking on the column headers in each window, you can sort the events by the appropriate field.*

High Priority Window

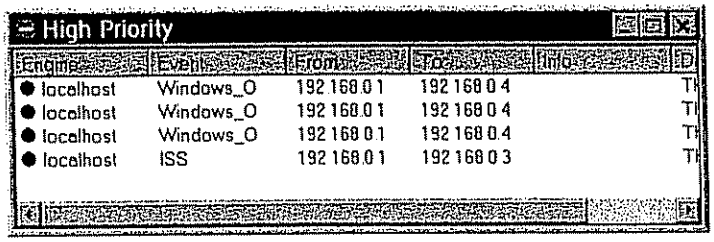


Figure 28: High Priority Window

This window displays the network events that are configured as "High Priority". Attacks are generally considered High Priority events To sort the entries, click the appropriate column header.

Medium Priority Window

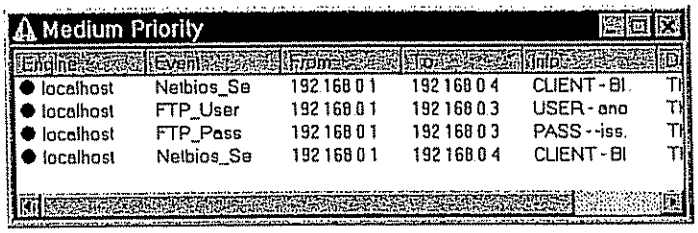


Figure 29: Medium Priority Window

This window displays the events that are configured "Medium Priority". Session decodes are generally considered Medium Priority events To sort the entries, click the appropriate column header.

Low Priority Window

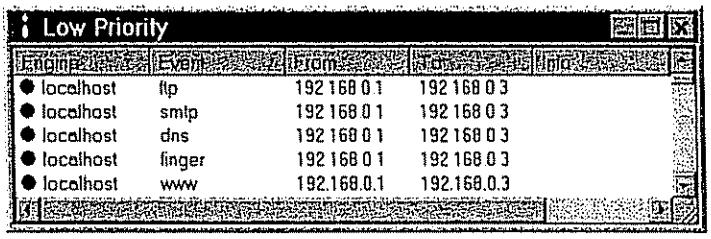


Figure 30: Low Priority Window

This window displays the events that are configured as "Low Priority". Filter events are generally considered to be Low Priority events. To sort the entries, click the appropriate column header.

The Event Inspector

You can examine the data that is displayed in RealSecure's Activity and Priority windows with the Event Inspector. To open the Event Inspector, right click in the Activity Tree (and only on the "leaf nodes").

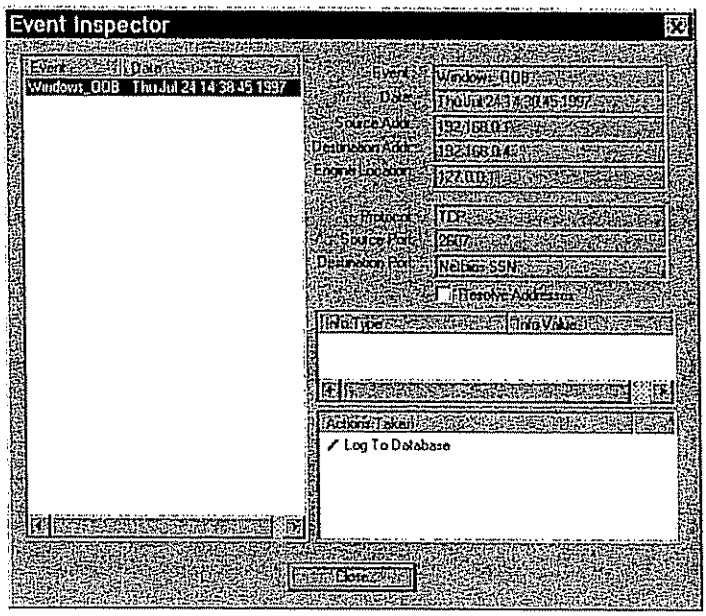


Figure 31: Event Inspector

The Event Inspector is shown reporting a Windows Out Of Band (OOB) Event.

Using the "Event Inspector, you can examine the data displayed in RealSecure's Activity, Priority and Session Playback windows To open it, right click in the Activity Tree (and only on the "leaf nodes").

The Event Inspector organizes the data gathered by RealSecure's engine in an easy-to-read format. The following information is displayed:

Event: The name of the reported network event. For a description, refer to Appendix A, Features and Attack Signatures.

Date: The date and time at which the event took place

Source Address: The IP Address of the machine which initiated the event.

Destination Address: The IP Address of the machine on which the event occurred.

Engine Location: The IP Address of the reporting Engine.

Protocol: The protocol associated with the Event.

Source Port: The source port of the Event.

Destination Port: The destination port of the Event.

Resolve Addresses: Event information is normally displayed using numeric IP addresses. If you click in the "Resolve Addresses" box, then the resolved domain or NetBIOS names will be used instead.

Info Type: If there is additional information associated with the Event, then this field will display the type of this information. For example: the piece of data associated with an HTTP_Get Event would be the URL that was retrieved.

Info Value: If there is additional information associated with the Event, then this field will display the value of this information For example: the piece of data associated with an HTTP_Get Event would be the URL that was retrieved.

Actions Taken: The box lists the Actions, if any, which the Console is configured to perform when this event is detected. For more information, see Actions.

Session Playback Window

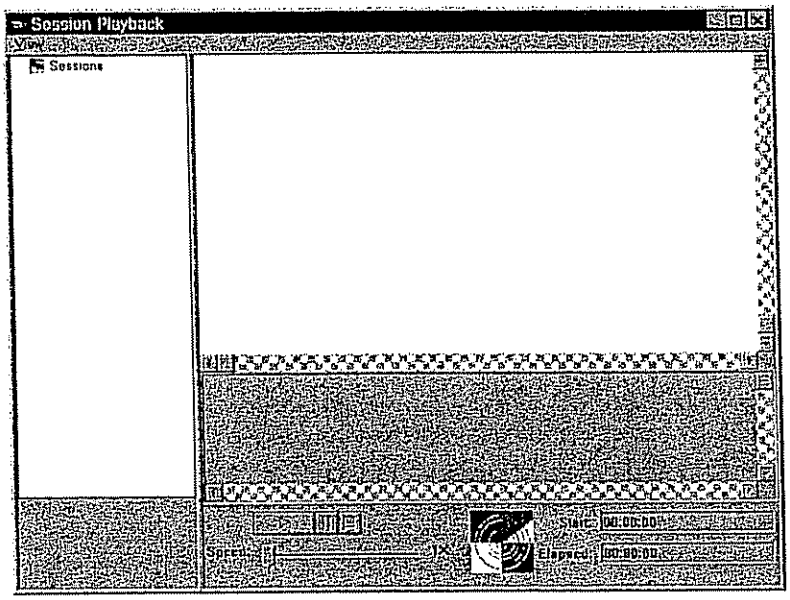


Figure 32: Session Playback Window

To open the Session Playback window, select "Session Playback" from the Console's View menu, or click the "View Session Playback Window" Toolbar button. The Session Playback Window is displayed.

The Session Playback window displays information logged during the monitoring session in two forms:

- **View binary information that is logged to the database:** The raw, binary content of an entire network session can be recorded. This data is stored in the engine's log database and can be replayed through the Management Console interface after you have synchronized all logs. It is played back exactly as it was received, keystroke for keystroke, so that you can see how the attack or session unfolded.

  Note: *You must configure the Engine to use the "Log Raw Data" option in order to play back logged data from the database.*

- **Play back an active session for the purpose of watching the session in real-time:** Sessions can be played back actively, as they happen.

  Note: *You must configure the Engine to use the "View Session" action in order to play back sessions in real time.*

To play back logged data, you must:

- Tell the Engine to "Log Raw Data".
- Upload the database from the Engine to the Console.

Saved sessions will appear in the "Logged Sessions" section of the session tree.

To view sessions in real-time you must tell the Engine to "View Session".

Sessions will appear in the "Active Sessions" section of the session tree.

RealSecure™

Chapter 5: Logging and Reporting Options in RealSecure for Windows NT

RealSecure for Windows NT logs database information both for Engines and for the Console. Database options for your Engines are configured through the Maintain Engine Log dialog box.

Maintain Engine Log Dialog Box

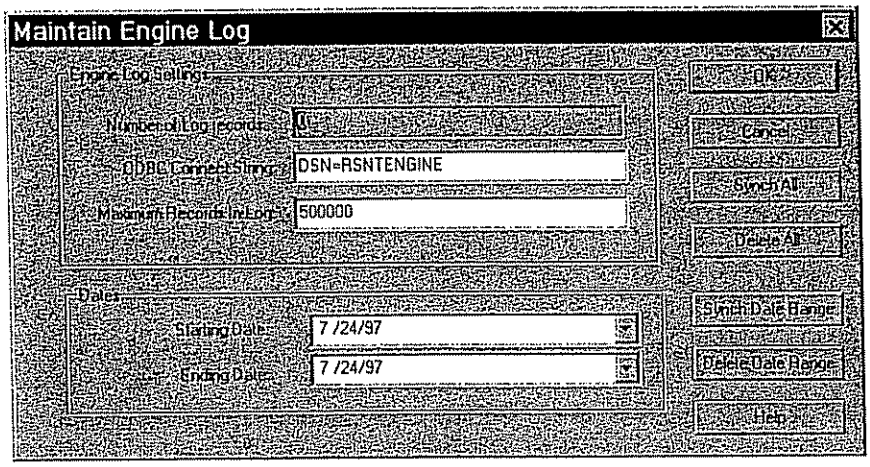


Figure 33: Maintain Engine Log dialog box

This dialog box allows you to configure the Engine Log for reporting purposes. To access this dialog, select "Maintain Engine Log" from the Console's View menu. The following parameters are configurable:

Engine Log Settings

Number of Log records: This value represents the number of log records presently stored in the Engine's log database

ODBC Connect String: This is termed as ODBC DSN and is a pointer to the file containing the database. This example shows that the Engine database is stored in the DSN named RSNT ENGINE LOG. You can use an ODBC applet within the Control Panel to reconfigure this if needed.

Maximum Records in Log: The maximum number of log records to store in the Engine Database. When this number is reached, the Engine overwrites the oldest record when a new record is added.

Dates

Starting Date: Type the date from which records should be copied for upload, or click the down arrow to select this date from the calendar. Used only for the "Synch Date Range" and "Delete Date Range".

Ending Date: Type the date through which records should be copied for upload, or click the down arrow to select this date from the calendar. Used only for the "Synch Date Range" and "Delete Date Range".

Buttons

OK: Saves changes and exits the dialog box.

Cancel: Exits without saving changes.

Synch All: Transfers all records from the selected Engine database to the Console's log database. The records are deleted from the Engine's log as they are transferred.

Delete All: Deletes all records in the Engine's log database.

Synch Date Range: Transfers records from the Engine's log database to the Console's log database, within the dates specified in Starting Date and Ending Date.

Delete Date Range: Deletes all records within the dates specified in Starting Date and Ending Date from the Engine's log database.

Help: Displays the On-line Help topic for this tab.

Maintain Console Log Dialog Box

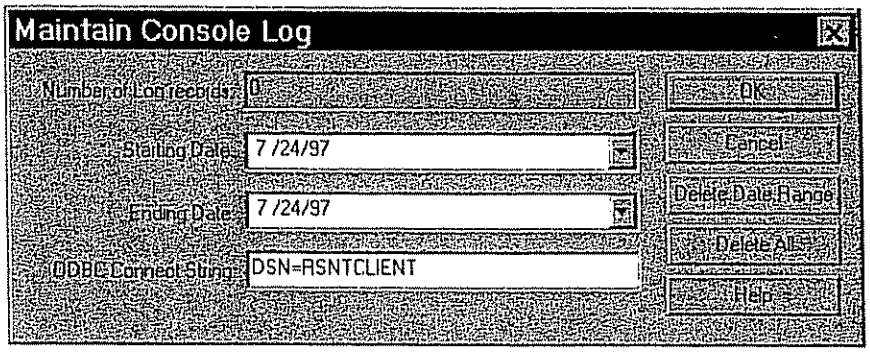


Figure 34: Maintain Console Log dialog box

This dialog box allows the administrator to configure the Console Log for reporting purposes. To access this dialog box, select "Maintain Log" from the File menu. The following parameters are configurable:

Number of Log Records: The number of log records presently stored in the Console's log database.

Starting Date: Type the starting date or click the down arrow to select from the calendar.

Ending Date: Type the ending date or click the down arrow to select from the calendar.

ODBC Connect String: A This is termed as ODBC DSN and is a pointer to the file containing the database. This example shows that the Engine database is stored in the DSN named RSNT ENGINE LOG. You can use an ODBC applet within the Control Panel to reconfigure this if needed.

Buttons

OK: Saves changes and exits the dialog box.

Cancel: Exits without saving changes.

Delete Date Range: Deletes all records within the dates specified in Starting Date and Ending Date from the Console's log database

Delete All: Deletes all records in the Console's log.

Help: Displays the On-line Help topic for this tab.

Reports Dialog Box

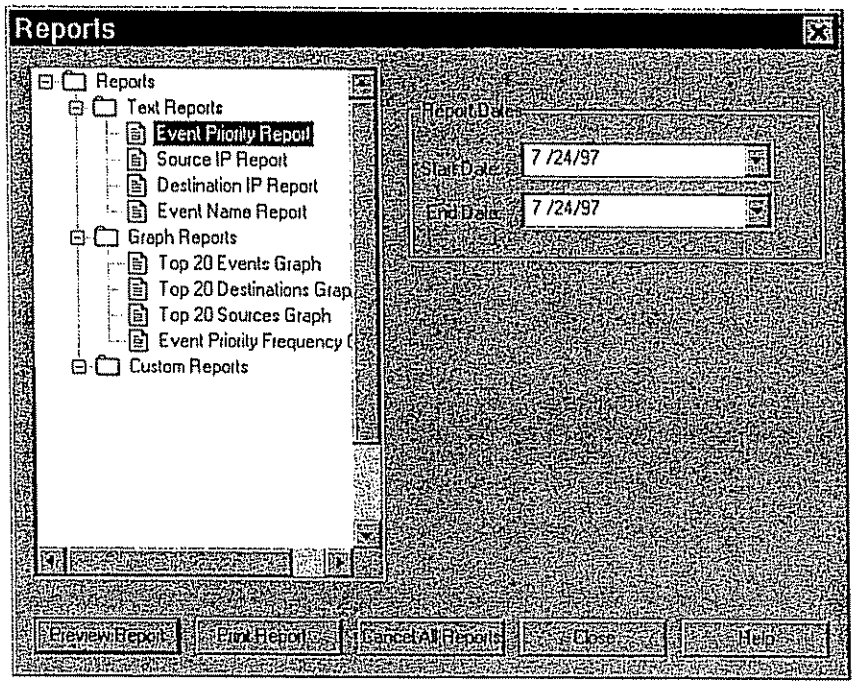
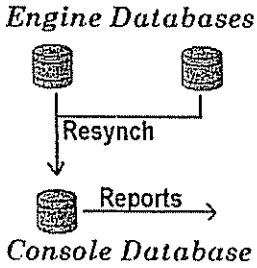


Figure 35: Reports dialog box

Reporting

Built-in report generation allows an administrator to get rapid, formatted summaries of network activity. RealSecure provides the ability to generate text-based and graphical activity reports. These reports essentially record date, time, source, and target of attacks. Report generation provides administrators the ability to tell where an attack came from and when it occurred. This information can be used to refine the network's defenses to prevent future attacks. These reports can be used later for the purpose of collecting evidence for future prosecution.

Note: *Reports are run against the Console's database, not the Engine's. A "Synchronize All Logs" is necessary to in order to pull out the latest engine database data to use in a report.*



Network events are stored in a log database for retrieval and playback. The Reports dialog box displays the reports available from within RealSecure. Twelve reports are provided, and the administrator can design custom reports as well. To open this dialog box, select "Reports" from the Console's View menu, or click the "View Reports" Toolbar button. Double-click on any of the report options to see a preview of how the selected report looks when printed.

There are three types of reports available, they are:

- Text
- Graph
- Custom

The following options are available:

Text Reports

Detailed reports containing all of the information associated with Events.

Event Priority Report: Lists details of all events over the specified time period sorted by priority first, and time second.

Source IP Report: Lists details of all events sorted first by source IP, then by time.

Destination IP Report: Lists details of all events sorted first by destination IP, then by time.

Event Name Report: Lists details of all events sorted by event name.

Graph Reports

These display summary counts of events in bar-chart form over the specified period of time.

Top 20 Events Graph: Graphs the top 20 event occurrences.

Top 20 Destinations Graph: Graphs the top 20 IP destinations, ranked by the number of events associated with each address.

Top 20 Sources Graph: Graphs the top 20 IP sources, ranked by the number of events associated with each address

Event Priority Frequency Graph: Graphs the number of high, medium, and low risk attack signatures were detected over the specified time frame.

Port Reports

Detailed reports containing all of the information associated with Events including Port information.

Event Priority Report: Lists details of all events over the specified time period sorted by priority first, and time second.

Source IP Report: Lists details of all events sorted first by source IP, then by time.

Destination IP Report: Lists details of all events sorted first by destination IP, then by time.

Event Name Report: Lists details of all events sorted by event name.

Custom Reports

Custom Reports: You can create your own custom reports and store them in the directory specified in the "Console Configuration" dialog box. This allows you to generate custom reports without ever having to exit RealSecure.

Report Dates

Start Date: Enter the date on which you want the report to begin. You can type the date or click the down arrow to select the start date from a calendar.

End Date: Enter the date on which you want the report to end. You can type the date or click the down arrow to select the end date from a calendar.

Buttons

Preview Report: Displays the report's contents in a preview window. For more information, see "Reports Viewer".

Print Report: Prints the selected report without previewing it.

Close: Closes the "Reports" dialog box.

Help: Click this button to view an On-line Help topic for this tab.

Reports Viewer

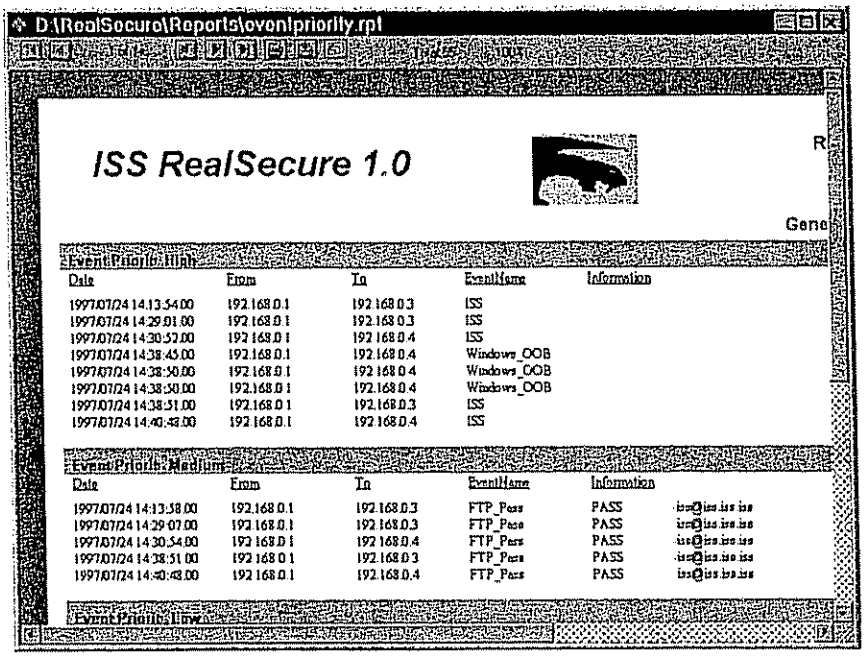


Figure 36: Reports Viewer

The Reports Viewer displays a preview of how the printed report will look. The Viewer is shown displaying an "Event Priority" report.

Note: *The path to the Reports folder is shown in the title bar. To open the "Reports Viewer", double-click on the desired report in the "Reports" dialog box.*

The following buttons allow for easier navigation through the Reports Viewer:


Click this button to go to the first page of the report.


Click this button to go back one page.


Click this button to close the report.


Click this button to go forward one page.


Click this button to go to the last page of the report


Click this button to print the report to a designated network printer.


Click this button to export the report to another application.


Click this button to zoom (in or out) the current report page from within the window.

RealSecure™

Appendix A: Attack Signatures

This appendix summarizes RealSecure's Features and the Attack Signatures it checks

- ARP Check
- Ascend Kill Denial of Service Vulnerability Check
- BootParamd Whoami Decode
- Chargen Denial of Service Vulnerability Check
- DNS Hostname Overflow Vulnerability Check
- DNS Length Overflow Vulnerability Check
- Echo Denial of Service Vulnerability Check
- E-Mail DEBUG Vulnerability Check
- E-Mail Decode Vulnerability Check
- E-Mail EXPN
- E-Mail From
- E-Mail Listserv Vulnerability Check
- E-Mail Pipe Vulnerability Check
- E-Mail Qmail Length Denial of Service Vulnerability Check
- E-Mail Qmail Rcpt Denial of Service Vulnerability Check
- E-Mail Subject
- E-Mail To
- E-Mail VRFY
- E-Mail WIZ Vulnerability Check
- Finger Bomb Vulnerability Check
- Finger User Decode
- FTP CWD ~root Vulnerability Check
- FTP get File Decoding
- FTP mkdir Decoding
- FTP Password Decoding
- FTP put File Decoding
- FTP Site Command Decoding

- FTP Site Exec Tar Vulnerability Check
- FTP Site Exec.. Vulnerability Check
- FTP Username Decoding
- HP/UX RemoteWatch Vulnerability Check
- HTTP ..Vulnerability Check
- HTTP Authentication Decode
- HTTP Campas Cgi-Bin Vulnerability Check
- HTTP GET Decoding
- HTTP Glimpse Cgi-Bin Vulnerability Check
- HTTP IIS 3 0 Asp 2E Vulnerability Check
- HTTP IIS 3 0 Asp Dot Vulnerability Check
- HTTP Internet Explorer 3 0 .URL/.LNK Vulnerability Check
- HTTP Java Decoding
- HTTP NCSA Buffer Overflow Vulnerability Check
- HTTP Novell Convert Vulnerability Check
- HTTP Nph-Test-CGI Vulnerability Check
- HTTP PHF Vulnerability Check
- HTTP PHP Buffer Overflow Vulnerability Check
- HTTP PHP File Read Vulnerability Check
- HTTP SCO View-Source Vulnerability Check
- HTTP SGI Wrap Vulnerability Check
- HTTP Test-CGI Vulnerability Check
- Ident Buffer Overflow Vulnerability Check
- Ident Newline Vulnerability Check
- Ident User Decoding
- IMAP Buffer Overflow Vulnerability Check
- IMAP Password Decoding
- IMAP Username Decoding
- INN Buffer Overflow Vulnerability Check
- INN Control Message Vulnerability Check
- IP Duplicate Check
- IP Fragmentation
- IP Half Scan
- IP Unknown Protocol
- IRC Channel Decode

- IRC Message Decode
- IRC Nick Decode
- IRCD Buffer Overflow Vulnerability Check
- ISS Scan Check
- Kerberos IV User Snarf Vulnerability Check
- Mountd Export Decode
- Mountd Mount Decode
- NETBIOS Session Grant Decode
- NETBIOS Session Reject Decode
- NETBIOS Session Request Decode
- NFS Guess Check
- NFS Mknod Check
- NFS UID Check
- NNTP Group Decoding
- NNTP Password Decoding
- NNTP Username Decoding
- PCNFSD Exec Vulnerability Check
- Ping Flooding
- Ping of Death Denial of Service Vulnerability Check
- POP Buffer Overflow
- POP Password Decoding
- POP Username Decoding
- Portmapper Program Dump Decode
- Portmapper Proxy Call Decode
- Portmapper Proxy Mount Check
- Portscan Detection Vulnerability Check
- RealSecure Kill Action Detection Check
- Rexec Session Decode
- Rlogin Decoding
- Rlogin -froot Vulnerability Check
- RPC Admind Check
- RSH Decoding
- RTM Finger Vulnerability Check
- Rwhod Vulnerability Check
- SATAN Vulnerability Check

- Selection Service Holdfile Check
- Source Routing
- SYN Flood
- Talk Flash Vulnerability Check
- Talk Request Decoding
- TFTP get Vulnerability Check
- TFTP put Vulnerability Check
- UDP Bomb
- Windows Null Session Decode
- Windows Out of Band Vulnerability Check
- Windows Password Cache File Access Vulnerability Check
- Windows Remote Registry Access Code
- Ypupdated Exec Check

ARP Check

ARP, Address Resolution Protocol, is used to determine the Ethernet address of a machine on a network given its IP address. If an ARP is received for a machine on the network, it immediately sends a reply. If the machine the ARP is destined for has crashed or otherwise disconnected from the network, several ARPs will be sent to it without any response This lack of response to ARP packets is used to determine if a machine on the network has crashed.

Ascend Kill Denial of Service Vulnerability Check

(requires filter for TCP port 23)

By sending a specially formatted malformed TCP packet to Ascend routers containing certain versions of the Ascend operating system, the router can be forced to cause an internal error, resulting in the router rebooting.

BootParamd Whoami Decode

Bootparamd is an RPC program used to facilitate diskless booting. An attacker trying to obtain a machine's NIS domainname can query Bootparamd's Whoami procedure for the domainname. Knowing the domainname allows the attacker to mount more NIS based attacks.

Chargen Denial of Service Vulnerability Check

(requires filter for UDP port 19)

This check watches for attempts at performing a denial of service attack against a machine on the network by attempting to engage a machine in a chargen flood against itself.

DNS Hostname Overflow Vulnerability Check

(requires filter for TCP and/or UDP port 53)

DNS responses for hostnames should not exceed a certain fixed length. Some versions of BIND do not validate this length, and hostnames longer than this length can be returned to programs doing DNS lookups. Programs that do not check the length of the hostnames returned may overflow internal buffers when copying this hostname, allowing a remote attacker to execute arbitrary commands on a targeted machine.

DNS Length Overflow Vulnerability Check

(requires filter for TCP and/or UDP port 53)

DNS responses for IP addresses contain a length field, which for all normal cases of IPv4 should be 4 bytes. By formatting a DNS response with a larger value than 4, certain programs executing DNS lookups will overflow internal buffers, allowing a remote attacker to execute arbitrary commands on a targeted machine

Echo Denial of Service Vulnerability Check

(requires filter for UDP port 7)

This check watches for attempts at performing a denial of service attack against a machine on the network by attempting to engage a machine in an echo flood against itself.

E-Mail DEBUG Vulnerability Check

(requires filter for TCP port 25)

The DEBUG command in Sendmail existed to allow debugging of a remote Sendmail daemon. It is no longer present in current versions of Sendmail, but old versions still in use allow an attacker to gain root access to a machine by using this command remotely.

E-Mail Decode Vulnerability Check

(requires filter for TCP port 25)

By sending mail to a decode or uudecode alias that is present in some systems, a remote attacker may be able to create or overwrite files on the remote host.

E-Mail EXPN

(requires filter for TCP port 25)

The EXPN command is used to expand the address of a user on a remote system. This is sometimes used legitimately to determine the full address of an intended mail recipient. It is also sometimes used to gain information about users on a system by trying to find out if certain common account names exist on a machine.

E-Mail From

(requires filter for TCP port 25)

This decoding discovers the sender of all mail that is sent over the network using SMTP.

E-Mail Listserv Vulnerability Check

(requires filter for TCP port 25)

This check recognizes a buffer overflow attack against the listserv mailing list management software. By sending a specific command through e-mail to the listserv software, an internal buffer in the program can be overflowed and arbitrary bytecode can be executed on the machine listserv is running on.

E-Mail Pipe Vulnerability Check

(requires filter for TCP port 25)

By inserting a pipe (|) character into certain fields in an e-mail, Sendmail may be forced to execute a command on the remote host. This results in a remote attacker being able to execute commands as root on the machine.

E-Mail Qmail Length Denial of Service Vulnerability Check

(requires filter for TCP port 25)

This check recognizes a Denial of Service attack against a Qmail mail server which sends a command string with an extensive length, causing Qmail to use all available RAM on the server it is running on.

E-Mail Qmail Rcpt Denial of Service Vulnerability Check

(requires filter for TCP port 25)

This check recognizes a Denial of Service attack against a Qmail mail server that is caused by repetitively RCPT commands to the server. An advanced parameter 'Email_Qmail_Rcpt_Threshold' can be configured to adjust the number of Rcpts that are legitimately allowed in a session before triggering this as an exploit. The default value for this parameter is 65535

E-Mail Subject

(requires filter for TCP port 25)

This decoding discovers the subject line of all mail that is sent over the network using SMTP.

E-Mail To

(requires filter for TCP port 25)

This decoding discovers the recipient of all mail that is sent over the network using SMTP.

E-Mail VRFY

(requires filter for TCP port 25)

The VRFY command is used to verify if a user on a remote system exists. This is sometimes used legitimately to determine if the recipient of a message at the intended destination is able to receive the message. It is also sometimes used to gain information about users on a system by attempting to determine if certain common account names exist on a machine.

E-Mail WIZ Vulnerability Check

(requires filter for TCP port 25)

The WIZ command in Sendmail existed to allow access to a machine under certain circumstances. It is no longer present in current versions of Sendmail, but old versions still in use may allow an attacker to gain root access to a machine by using this command.

Finger Bomb Vulnerability Check

(requires filter for TCP port 79)

This check watches for attempts to perform a denial of service attack against a machine or for redirecting finger attempts across machines. Redirecting finger attempts is often used by an attacker to hide the original source address of a finger attempt.

Finger User Decode

(requires filter for TCP port 79)

This decode watches for finger attempts and reports the user (or all users if the attempt was aimed at the whole machine) that the finger was aimed at. Finger has a legitimate use, but is also often used by attackers to gain more information about a machine such as account names, real names, and trusted hosts.

FTP CWD ~root Vulnerability Check

(requires filter for TCP port 21)

Certain versions of the FTP daemon allow access to files on a machine through a sequence of commands culminating with CWD ~root. This vulnerability allows an attacker who can access FTP on the target host to transfer files to which he/she would not normally have access.

FTP GET File Decoding

(requires filter for TCP port 21)

Files being transferred from the destination host to the source host use a GET command in order to transfer the files. FTP GET decoding discovers all files that are being transferred to the source host over FTP.

FTP Mkdir Decoding

(requires filter for TCP port 21)

FTP allows a user to create a new directory on the target machine. FTP Mkdir decoding discovers all new directories that are created through FTP.

FTP Password Decoding

(requires filter for TCP port 21)

FTP passes a plain text password across the network in order to authenticate that a user has access to the files on the destination host. This password is discovered using FTP password decoding. This allows an administrator to log invalid password attempts, check passwords for strength against attack and keep complete logs of activity.

FTP PUT File Decoding

(requires filter for TCP port 21)

Files being transferred from the source host to the destination host use a PUT command in order to transfer the files. FTP PUT decoding discovers all files that are being transferred to the destination host over FTP.

FTP Site Command Decoding

(requires filter for TCP port 21)

The FTP site command allows a user to execute certain commands on a destination host in addition to the normal FTP facility of transferring files. In ordinary usage of FTP, this is not a commonly used command. While there may be a legitimate reason to execute site commands under certain circumstances, this facility has also been used to gain access. Consequently, an administrator may wish to view and log the site commands being executed to check for possible abuse

FTP Site Exec Tar Vulnerability Check

(requires filter for TCP port 21)

Certain versions of wu-ftpd allow using a site exec command to execute commands on a remote machine. A command line option to the GNU tar program allows a user with FTP access to execute arbitrary commands on an ftp server.

FTP Site Exec.. Vulnerability Check

Certain versions of wu-ftpd allow using a site exec command to execute commands on a remote machine. By providing a pathname with certain characteristics, a remote user can execute arbitrary commands on the ftp server.

FTP Username Decoding

(requires filter for TCP port 21)

FTP, File Transfer Protocol, allows users to transfer files between machines. Username decoding discovers the name of the account being used to transfer files across the network.

HP/UX RemoteWatch Vulnerability Check

(requires filter for TCP port 5556)

Certain versions of HP/UX that come with the RemoteWatch package installed have a vulnerability which allows a remote attacker to execute arbitrary commands through the RemoteWatch service on the target machine. This vulnerability check will watch accesses to the RemoteWatch service and determine if these accesses are attempting to exploit his vulnerability.

HTTP ..Vulnerability Check

(requires filter for TCP port 80)

This check recognizes an attack to attempt to obtain information above the "ServerRoot" directory. Some web servers vulnerable to this attack will allow remote users to list the contents of any directory on the system using this type of attack.

HTTP Authentication Decode

(requires filter for TCP port 80)

This decode logs the username and password used to authenticate HTTP Basic authentication to a web server. This authentication uses Base64 encoding and can be used for such purposes as determining what user accounts are logging into web servers from what machines, log brute-force attacks against the web server, and to keep general logs of username and password attempts.

HTTP Campas Cgi-Bin Vulnerability Check

(requires filter for TCP port 80)

This check recognizes an attack against the campas cgi-bin script present with certain HTTPD web servers. This exploit allows a remote attacker to execute commands on the web server machine as the user the HTTPD process is running as.

HTTP GET Decoding

(requires filter for TCP port 80)

Pages, images, and all other information viewed through a Web browser on the World Wide Web are transferred through HTTP using the GET command. HTTP GET decoding discovers all Web pages being transmitted unsecurely to a machine. This allows an administrator to track, log and view Web traffic on the network.

HTTP Glimpse Cgi-Bin Vulnerability Check

(requires filter for TCP port 80)

This check will recognize an attack against the glimpse cgi-bin script present with certain HTTPD web servers. This exploit allows a remote attacker to execute commands on the web server machine as the user the HTTPD process is running as.

HTTP IIS 3.0 Asp 2E Vulnerability Check

(requires filter for TCP port 80)

Microsoft's IIS 3.0 server installed with the hot-fix to solve the ASP Dot vulnerability introduced a new security hole that allows viewing the contents of an active server push URL by using the hexadecimal value '2e' instead of a '.' in the URL name. This check will recognize attempts to exploit this vulnerability to view the contents of pages.

HTTP IIS 3.0 Asp Dot Vulnerability Check

(requires filter for TCP port 80)

Microsoft's IIS 3.0 server has a security hole that allows execution of code by inserting a '.' after an active server push URL. This check will recognize attempts to exploit this vulnerability.

HTTP Internet Explorer 3.0 .URL/.LNK Vulnerability Check

(requires filter for TCP port 80)

Microsoft's Internet Explorer versions 3.0 and 3.01 have a vulnerability which results in a web site being able to execute an arbitrary program on a machine running Microsoft Windows and browsing the web using MSIE. This vulnerability check detects when a web site attempts to exploit this vulnerability.

HTTP Java Decoding

(requires filter for TCP port 80)

This decoding recognizes when a web browser attempts to obtain a file containing Java bytecode. This should only occur if a user has Java enabled on their web browser.

HTTP NCSA Buffer Overflow Vulnerability Check

(requires filter for TCP port 80)

This check recognizes a buffer overflow attack against certain versions of the NCSA HTTPD web server. This exploit allows a remote attacker to execute arbitrary code on the web server machine as the user the HTTPD process is running as.

HTTP Novell Convert Vulnerability Check

(requires filter for TCP port 80)

This check recognizes an attack on the convert bas cgi-bin program included as part of some versions of Novell's HTTP server. By accessing the convert program with specially formatted arguments, a remote attacker can view the contents of any file on the system with read permissions by the process the web server is running as.

HTTP Nph-Test-CGI Vulnerability Check

(requires filter for TCP port 80)

This check recognizes an attack on the cgi-bin nph-test-cgi script. This program, installed by default with certain versions of Apache and NCSA web servers, allows remote attackers to gain information about the contents of the cgi-bin directory of the web server which can be used for further attacks.

HTTP PHF Vulnerability Check

(requires filter for TCP port 80)

The cgi-bin script PHF, which comes pre-installed with several versions of NCSA and Apache Web servers, contains a vulnerability that allows anyone who can access a Web site to the machine(s).

HTTP PHP Buffer Overflow Vulnerability Check

(requires filter for TCP port 80)

This check recognizes an attack on the PHP cgi-bin program. By overflowing a buffer in the PHP program, a remote attacker can execute commands as the user the HTTPD process runs as on a web server.

HTTP PHP File Read Vulnerability Check

(requires filter for TCP port 80)

This check recognizes an attack on the PHP cgi-bin program. By accessing the php cgi program with specially formatted arguments, a remote attacker can obtain directory listings of directories on the web server, providing the attacker with information about the machine.

HTTP SCO View-Source Vulnerability Check

(requires filter for TCP port 80)

This check recognizes an attack on the view-source cgi-bin script included as part of SCO Skunkware CD-ROM distributions and other HTTP servers. By accessing the view-source script with specially formatted arguments, a remote attacker can view the contents of any file on the system with read permissions by the process the web server is running as.

HTTP SGI Wrap Vulnerability Check

(requires filter for TCP port 80)

This check recognizes an attack on the wrap cgi-bin script included as part of the WWW HTTP server shipped, with IRIX 6.2. By accessing the wrap script with specially formatted arguments, a remote attacker can obtain directory listings of directories on the web server, providing the attacker with information about the machine.

HTTP Test-CGI Vulnerability Check

(requires filter for TCP port 80)

This check recognizes an attack on the cgi-bin test-cgi script. This program, installed by default with certain versions of Apache and NCSA web servers, allows a remote attacker to gain information about the contents of the cgi-bin directory of the web server which can be used for further attacks.

Ident Buffer Overflow Vulnerability Check

(requires filter for TCP port 113)

Certain programs that connect back to the ident service to log user information, expect a properly formatted response. If the response is longer than expected, the buffer that the response is read into is overflowed, allowing the remote user to execute commands on the host machine.

Ident Newline Vulnerability Check

(requires filter for TCP port 113)

Certain programs that connect back to the ident service to log user information expect a properly formatted response. If the response contains newlines, the response may be improperly parsed, allowing the remote user to execute commands on the host machine.

Ident User Decoding

(requires filter for TCP port 113)

The Ident port is used by services to identify the account by which a connection is being made on a machine. This can be used to track a connection back to a specific user on a multi-user machine.

IMAP Buffer Overflow Vulnerability Check

(requires filter for TCP port 143 and/or 220)

Certain versions of IMAP mail servers contain a vulnerability that allows a remote attacker to gain root access to a machine by overflowing an internal buffer in the IMAP server.

IMAP Password Decoding

(requires filter for TCP port 143 and/or 220)

The IMAP service is used by numerous e-mail programs to retrieve e-mail from a mail server and read it on a local machine. IMAP password decoding discovers all successful passwords that a user attempts to use to login to a mail server using IMAP.

IMAP Username Decoding

(requires filter for TCP port 143 and/or 220)

The IMAP service is used by numerous e-mail programs to retrieve e-mail from a mail server and read it on a local machine. IMAP password decoding discovers all successful and unsuccessful passwords that a user attempts to use to login to a mail server using IMAP.

INN Buffer Overflow Vulnerability Check

(requires filter for TCP port 119)

This check recognizes a buffer overflow attack against the INN news server. This exploit allows a remote attacker by sending a specially crafted message through the nnrpd process to overflow a buffer in the INN news server and execute arbitrary bytecode on the machine running the INN server.

INN Control Message Vulnerability Check

(requires filter for TCP port 119)

This check recognizes an attack against the INN news server that allows any remote user who can propagate a message to the news server to execute arbitrary commands on the remote machine.

IP Duplicate Check

Only one machine on a network should send packets with a specific IP address. If a second machine on the network starts to send packets claiming to have the same source address, a network problem has occurred. A machine on the network may be misconfigured to have the same IP address as another machine, causing network conflicts. The other possibility is that a machine on the network may be sending out IP packets with a forged source address.

IP Fragmentation

An IP packet is sometimes split into several fragments when it is transmitted over the network. These fragments are then reassembled at the destination to form a full IP packet. Some routers that filter out packets based on information in the TCP header rely on the information in the first fragment, then blindly pass the remaining fragments. It is possible to construct individual fragments of an IP packet so that subsequent packets overlap. As a result, parts of the TCP header are overwritten when they are reassembled at the destination. The result of this is that an intermediate filtering router is tricked into believing that a packet is destined for an allowed service. In reality, the packet is destined for a service that would normally be filtered.

IP Half Scan

A standard TCP connection is established by sending a SYN packet to the destination host. If the destination is waiting for a connection on the specified port, it will respond with a SYN/ACK packet. The initial sender then replies to the SYN/ACK with an ACK packet, and the connection is established. If the destination host is not waiting for a connection on the specified port, it will respond with an RST packet instead of a SYN/ACK. Most system logs do not log completed connections until the final ACK packet is received from the source. Sending an RST packet instead of the final ACK results in the connection never actually being established; so no logging takes place. Because the source can identify whether the destination host sent a SYN/ACK or an RST, an attacker can determine exactly what ports are open for connections, without the destination ever being aware of probing.

IP Unknown Protocol

A standard IP packet contains an 8-bit protocol field. Common values for this field include 6 (TCP), 17 (UDP), and 1 (ICMP). Attackers sometimes use a non-standard value for this field, in order to exchange data between machines without logging mechanisms detecting the data that is being transmitted.

IRC Channel Decode

(requires filter for TCP port 6667)

This decode watches for channels that are joined by a user on Internet Relay Chat.

IRC Message Decode

(requires filter for TCP port 6667)

This decode watches for messages that are sent out by a user on Internet Relay Chat.

IRC Nick Decode

(requires filter for TCP port 6667)

This decode watches for changes of a user's nickname on Internet Relay Chat.

IRCD Buffer Overflow Vulnerability Check

(requires filter for TCP ports 6666-6669)

This check recognizes a buffer overflow attack against ircd, the server binary for Internet relay chat. This exploit allows a remote attacker to execute arbitrary bytecode on the machine running the ircd process.

ISS Scan Check

(Requires filter for ICMP Type = 8, Code = 0)

This check recognizes an ISS scan taking place. This recognizes vulnerability assessments being made with the freely available version of Internet Security Scanner, or with the commercial version of the product from Internet Security Systems, Inc.

Kerberos IV User Snarf Vulnerability Check

(requires filter for UDP port 750

Kerberos version 4 contains a vulnerability that allows a remote attacker to gain username and realm information from a kerberos server by passing a malformed packet to the server.

Mountd Export Decode

This decode detects a remote showmount.

Mountd Mount Decode

This decode detects an NFS mount request.

NetBIOS Session Grant Decode

(requires filter for TCP port 139)

This decode recognizes when a NetBIOS session that has an outstanding session request has been granted the session. This indicates a session has been successfully established between the two machines.

NetBIOS Session Reject Decode

(requires filter for TCP port 139)

This decode recognizes when a NetBIOS session that has an outstanding session request has been rejected. When available, a reason for the rejection will be provided. This indicates an attempted session has been denied between the two machines.

NetBIOS Session Request Decode

(requires filter for TCP port 139)

This decode recognizes when a NetBIOS session has been requested to be initiated between two machines. This check decodes the NetBIOS names of the source and destination machines.

NFS Guess Check

Most NFS implementations have specific patterns in their filehandles which can be guessed. Since most NFS implementations rely on the secrecy of the filehandle for the files actual security, an attacker can guess filehandles and access NFS resources with or without Authentication.

NFS Mknod Check

Some NFS implementations allow the nfsproc_create procedure to create special devices on the exported filesystem. If an attacker can create devices he can more often than not compromise the security of the site.

NFS UID Check

For security reasons most NFS implementations map the root user to the "nobody" user. Under the NFS protocol the UID is a 32-bit value, and under most versions of UNIX, the UID is a 16bit value. Therefore, an attacker can submit a non-zero 32-bit UID which will in actuality be treated as a zero 16-bit UID by the operating system.

NNTP Group Decoding

(requires filter for TCP port 119)

The NNTP service is used to read, post, and exchange news from a news server. NNTP group decoding discovers the name of the newsgroup that a user is accessing on the news server.

NNTP Password Decoding

(requires filter for TCP port 119)

The NNTP service is used to read, post, and exchange news from a news server. NNTP password decoding discovers the password attempted to login to the news server in order to read or post news.

NNTP Username Decoding

(requires filter for TCP port 119)

The NNTP service is used to read, post, and exchange news from a news server. NNTP user decoding discovers the username of the user who is reading or posting news through the NNTP service.

PCNFSD Exec Vulnerability Check

This check recognizes an attack against the PCNFSD service that allows a remote attacker to execute arbitrary commands on the machine being attacked.

Ping Flooding

A Ping Flood is an attempt to saturate a network with packets in order to slow or stop legitimate traffic going through the network. A continuous series of ICMP Echo Requests are made to a target host on the network, which then responds with an ICMP Echo Reply. The continuing combination of requests and replies slow the network and cause legitimate traffic to continue at a significantly reduced speed or, in extreme cases, to disconnect.

Ping of Death Denial of Service Vulnerability Check

This check recognizes an attempt to send an oversized ICMP ping packet to a host in order to execute a denial of service attack against the machine. Machines vulnerable to this attack will crash, reboot, or lose network connectivity when this packet is sent.

POP Buffer Overflow

(requires filter for TCP port 109 and/or 110)

Certain versions of POP mail servers contain a vulnerability that allows a remote attacker to gain root access to a machine by overflowing an internal buffer in the POP server.

POP Password Decoding

(requires filter for TCP port 119)

POP password decoding discovers all successful and unsuccessful passwords that a user attempts to use to login to a mail server using POP.

POP Username Decoding

(requires filter for TCP port 119)

The POP service is used by numerous e-mail programs to retrieve e-mail from a mail server and read it on a local machine. POP username decoding discovers the username of the user who is reading mail through the POP service.

Portmapper Program Dump Decode

This decode detects a remote listing of a machine's RPC programs.

Portmapper Proxy Call Decode

This decode detects a proxy procedure call through portmapper.

Portmapper Proxy Mount Check

This check detects someone attempting to mount NFS filesystems using portmapper's proxy service.

Portscan Detection Vulnerability Check

This check recognizes a portscan that is taking place on the network. A portscan is an attempt by an attacker to enumerate the services running on a machine by probing each port for a response. This vulnerability check will detect a normal portscan, as well as stealth scans (sometimes also referred to as Half Scans, SYN/ACK Scans, or FINscans). This check has two configurable parameters allowing you to customize the sensitivity of the detection. On typical networks, the default settings for these parameters do not need to be changed. The first parameter is Port_ScanPorts. This is the number of ports that need to be probed in order for this signature to be triggered. A smaller value will increase sensitivity and the speed at which a portscan can be detected, but can also result in false positives caused by normal network activity. The second parameter is Port_ScanDelta. This is the window of time, in seconds, that Port_ScanPorts ports need to be probed within in order to trigger the signature. A smaller value for this will conserve memory usage and reduce sensitivity to being scanned; a higher value will enable detecting a time delayed portscan, at the expense of memory consumption and the possibility of false positives.

RealSecure Kill Action Detection Check

This check recognizes TCP RST packets being sent from a RealSecure Engine on the network. This enables a RealSecure Engine to determine if other RealSecure Engines on a network are killing TCP connections, explaining losses in network connectivity depending on configurations. Newer versions of RealSecure (UNIX versions 1.2.2 and above, and all versions of NT RealSecure) also have the option to embed the customer ID in RST packets. If this option is set, this check extracts that customer ID from the RST and allow one to determine the source of the kill.

Rexec Session Decode

(requires filter for TCP port 512)

This decode determines when an rexec command has been executed and provides the details of the program being run. Rexec is a service used to execute commands on a remote machine.

Rlogin Decoding

(requires filter for TCP port 513)

An Rlogin connection allows a user to remotely login to a host without a password by using a trust relationship between the account on the source machine and on the destination host. The source machine and username, along with the destination machine and username are logged with this feature.

Rlogin -froot Vulnerability Check

(requires filter for TCP port 513)

If a remote user passes the name -froot to rlogin to a machine, certain operating systems will bypass normal security mechanisms and log in the user directly as root. This vulnerability allows anyone who can access the rlogin service on the target host to gain immediate root access to the machine.

RPC.Admind Check

RPC.Admind is used for remote administration of Solaris machines. When RPC Admind is used with insecure authentication, attackers can compromise the machine.

RSH Decoding

(requires filter for TCP port 512)

RSH, the remote shell command, allows a user to execute a shell command over the network using a trust relationship between the user on the local machine and the user account on the remote machine. RSH decoding discovers both the local and remote usernames as well as the command that is being executed.

RTM Finger Vulnerability Check

(requires filter for TCP port 79)

This check watches for a buffer overflow attempt on the finger service that is used by attackers to attempt to gain access to a machine remotely. This vulnerability is named for Robert T. Morris, author of the Internet Worm that originally popularized this vulnerability.

Rwhod Vulnerability Check

(requires filter for UDP port 513)

This check watches for a malformed rwho UDP packet containing a buffer overflow, that can be used by attackers to perform a denial of service attack against the rwho service or to attempt to execute arbitrary code on a remote machine.

Satan Vulnerability Check

(requires filter for UDP port 1)

This check will recognize if a Satan normal or heavy scan of a machine is taking place. Satan is a freely available tool that allows someone to scan a machine for services and a small set of common vulnerabilities.

Selection Service Holdfile Check

The selsvc RPC program is used by Suntool for, among other things, file access. There should not be any remote file access through selsvc. RealSecure will report any remote file access using selsvc.

Source Routing

IP packets sent over the Internet are normally sent between different routers, in order to reach their final destination. The route each packet takes is determined dynamically by each router along the way. Enabling the source routing option on an IP packet allows the packet itself to make known to each router, the path it wishes to take to reach its final destination. By routing packets through a path that bypasses filtering routers and other normal security mechanisms, an attacker may be able to reach a host that normally could not be reached. Also, it can be used to authenticate an intruder to systems that rely on the source IP address for access control.

SYN Flood

A standard TCP connection is established by sending a SYN packet to the destination host. If the destination is waiting for a connection on the specified port, it will respond with a SYN/ACK packet. The initial sender then replies to the SYN/ACK with an ACK packet, and the connection is established. When the SYN/ACK is sent back to the source, a block of memory is allocated to hold information about the state of the connection that is currently being established. Until the final ACK is received or a timeout is reached, this block of memory sits unused, waiting for more information to be received from the source host. By sending numerous SYN packets to a host, the destination will exhaust the portion of memory it has on-hand to deal with opening connections. Legitimate connections will no longer be able to connect to the host. This situation can be detected by the flood of SYN packets without accompanying responses. It can be corrected by sending the destination RST packets that correspond to the initial SYNs. This results in the destination host freeing up that block of memory and making room for a new legitimate connection.

Talk Flash Vulnerability Check

(requires filter for UDP port 517 and 518)

The talk service allows the user originating a talk request to specify an arbitrary string to display for the origin of the talk request. If this string contains a particular escape sequence, it is possible to cause a temporary denial of service attack by mangling the contents of a user's screen. This is commonly known as 'flashing' a user.

Talk Request Decoding

(requires filter for UDP port 517 and 518)

The Talk service is used to engage in a real-time chat with a user on a remote machine. Talk Request decoding discovers the name and machine that a talk request is being sent to, along with the name and machine of the person who is originating the talk request.

TFTP Get Vulnerability Check

(requires filter for UDP port 69)

This check watches for attempts to transfer files from a machine using the Trivial File Transfer Protocol (TFTP). This protocol is sometimes legitimately used for bootstrapping by diskless workstations, but it is more often used by attackers to attempt to obtain a password file or other critical system files.

TFTP Put Vulnerability Check

(requires filter for UDP port 69)

This check watches for attempts to transfer files to a machine using the Trivial File Transfer Protocol (TFTP). This protocol can be used by attackers to transfer critical system files to a host that is being attacked.

UDP Bomb

A UDP packet that is constructed with illegal values in certain fields will cause some older operating systems to crash when the packet is received. If the target machine does crash, it is often difficult to determine the cause. Most operating systems that are not vulnerable to this problem will silently discard the invalid packet, leaving no traces that it was being subjected to a malicious attack.

Windows Null Session Decode

(requires filter for TCP port 139)

This decode recognizes when a null user session has been established between two machines. This is an indication that two machines are communicating using the anonymous user. This can be a legitimate connection between two machines, or can be an indication of a "Red Button" null session attack.

Windows Out of Band Vulnerability Check

(requires filter for TCP port 139 and/or TCP port 53)

This check recognizes an out of band denial of service attack against a Windows machine. This attack causes a complete crash of a machine (blue screen) or loss of network connectivity on vulnerable machines. This attack, also known as "WinNuke" has two variations, an original WinNuke and a second attack known as "WinNuke 2" or "Mac WinNuke." Both of these attacks are recognized by this check.

Windows Password Cache File Access Vulnerability Check

(requires filter for TCP port 139)

This check recognizes an access of a PWL password cache file over a NetBIOS share. PWL cache files are weakly encrypted and accessing these files over a network can be an indication of an attacker attempting to retrieve these files. Even in cases of the legitimate user accessing his or her own cache file, it is sent unencrypted over the network and represents a risk.

Windows Remote Registry Access Code

(requires filter for TCP port 139)

This check recognizes an access of the registry on a remote machine over a NetBIOS session. The registry can be accessed remotely either through a registry modification tool such as regedit, or as an automated part of normal network activity.

Ypupdated Exec Check

This check detects someone attempting to gain unauthorized access to the machine using security problems in ypupdated.

RealSecure™

Appendix B: Troubleshooting

This appendix addresses some common questions and potential problems with using RealSecure for Windows NT. See also the readme file for the latest news.

My point-cast seems to set off the SYN flood signature detection check. What can I do?

Here is the problem. Pointcast, since it works over HTTP, treats every object as a separate connection. This in and of itself can easily set off a syn-flood alarm. The best thing that can be done is to go to the advanced parameters under synflood and increase the value for the SynFloodHighWaterMark variable. Basically, the higher this number is, the less chance of a false positive. However, this also can mean that a carefully crafted syn flood could escape through the check as most syn floods are high volume enough to trigger no matter what that variable is.

If I set up a remote Engine and monitor it remotely, will it send the database files to me as well?

Currently, RealSecure for Windows NT requires that each Engine reports to a single Console. However, database files remain on the Engine machine so that in the event of a loss of network connectivity between the Engine and Console machines, the logs remain fully intact. You can transfer database files back to the Console machine by using the transfer files option in the Engine maintenance dialog box.

Will the DNS resolutions that RealSecure does load my network?

No, the queries for DNS are usually cached for a period of time and the request does not have to be made again for subsequent look-ups for the same machine. However, if there is a problem with RealSecure and DNS and you wish to disable DNS resolution, edit the general.cfg file so that the line with 'RESOLVE' is set to 'no'.

When I run the multiple RealSecure Engines remotely and I monitor them from another supported platform, will that affect my network load?

No, it should not. The only traffic that is transmitted from the Engine running remotely is a single UDP packet for each reportable event as defined in the filter rule set and each reportable attack signature as defined in the "active" signature database of the respective Engine.

When I run a RealSecure Engine remotely, the Engine sometimes changes from a state of 'Active" to 'No Response' for a few seconds, and then changes back to an 'Active' state. Why does it do this?

A 'No Response' state is an indication that the Engine has not yet responded to a ping sent out from the GUI. This can be an indication that the Engine is busy doing other higher priority tasks and has not yet had time to ping back to the GUI. This could also occur if a machine or Engine has crashed, or lost connectivity with the network. If the Engine periodically changes to this state for a few seconds to a minute, the most likely cause of this is a non-responsive DNS server.

How does the Engine report data back to the Console? What protocols/services are involved if I am going to run the remote Engine on the outside of our firewall and need to communicate through our boundary host?

The RealSecure Engine (sitting outside of your firewall) communicates back to the RealSecure Console inside the firewall using two mechanisms. The first mechanism is event passing through UDP packets. The RealSecure Engine sends UDP packets back to the Console host destined for UDP port 900. In addition, the RealSecure Engine listens for UDP packets from the GUI on UDP port 901. If you have your firewall configured to allow all outgoing UDP packets but no incoming, you need to add a firewall rule to allow UDP packets from the RealSecure Engine IP, any source port, to the RealSecure Console IP, destination port 900. If you are blocking UDP traffic out of the network, you need to add a rule to allow outgoing UDP packets from the RealSecure Console IP, any source port, to the RealSecure Engine IP, destination port 901. Alternatively, you can change the UDP ports used by the Engine to communicate with the Console.

Using the IP and a shared secret pass-phrase based authentication, the RealSecure Console inside the firewall authenticates all packets coming from the Engine outside the firewall. Further, the Engine communicates information that it has gathered to the Console, but the Console does not allow the Engines to gather any information from the Console host, so the communication channel back through the firewall is a 1-way flow of security information. There is no way an Engine outside of a firewall could in any way compromise the RealSecure Console host or any other machine inside the firewall through this communications channel. If you are using an entirely proxy-based firewall, you can setup a generic UDP proxy to proxy this communication instead of directly passing the packets. If this is the case, you will need to change your RealSecure sss authentication entries to point to the proxy host (since this is the host the Engine and Console will see as the source address on the packets).

The second level of Console-Engine communication involves starting, stopping, configuration changes, and database uploads. This communication takes place over TCP with a destination port of 590. If your firewall allows arbitrary outgoing TCP connections, you do not need to make any modifications to use this. If the firewall blocks outgoing TCP connections, you need to add a firewall rule (or generic TCP proxy) allowing connections from the Console IP, any reserved port, to the Engine IP, destination port 590. Again, this communication is initiated and controlled solely by the Console host inside the firewall, and so it provides no possibility of gaining or compromising any machine inside the firewall through this mechanism.

I have configured a filter rule with the 'View' or 'Log Raw' action, but when I view the session, I only see data at the bottom of the screen, and nothing happens in the terminal window.

In order to view or log raw data for both directions in a connection, you need to create 2 filter rules. The first filter rule will tell RealSecure to view or log all data going TO a specific port. The second rule you add will view or log all the data that is coming FROM the port. If you only use one rule, then you will only view or log data in one direction for a connection, not the full-duplex connection. For example, to properly view all Telnet sessions in real-time, you should have the following 2 filter rules:

Source Address	Destination Address	Protocol	Source Port	Destination Port	Actions
*	*	TCP	Any	Telnet	V
*	*	TCP	Telnet	Any	V

Can I specify groups of IP addresses, ranges and the like in the "Source" and "Destination" fields in the "Filter Configuration"?

No. RealSecure reads the Source and Destinations as bit maps and hence will only recognize IP addresses.

What if I wanted to specify a range of IP addresses within the rule set? Is there a way to do that?

Yes, by using a series of rules that you can define in the filter set. Since the rules are read from the top down, the user can easily specify a set of rules that view a Telnet session of an IP address, and only display the events of all others. An example of this is below:

Source Address	Destination Address	Protocol	Source Port	Destination Port
192 01 01 01	Any	TCP	Any	Telnet
Any	192 01.01 01	TCP	Telnet	Any

Can I use Wild Cards in the Source and Destination fields?

As in the examples of rule-sets, you can use wild-card designations in the "Source" and "Destination" fields. Use these with care, however, as if they are associated with the "kill" action you could accidentally impact the functionality of your respective network. One more note about the use of wildcards. You can use wildcards in the GUI on octet boundaries EG, 10.0.* is legal, but 10.0.1* is not.

What types of attack events will RealSecure recognize?

Because the core competency of developers for Internet Scanner SAFEsuite was knowing what vulnerabilities exist and how to exploit them, that knowledge base was leveraged in building RealSecure to have the most comprehensive set of attack patterns that will catch all the known exploits. This is a continually evolving process and the database of detected signatures grows with every release.

The attacks range from low level IP attacks, such as IP Fragmentation (which may by-pass some packet filter firewalls) and SYN flooding attacks to high level attacks such as Web, FTP, NFS, NIS, and E-mail attacks. See Appendix A, "Attack Signatures," for more information on specific attacks.

How are the rules in the filter processed?

From top to bottom. If none of the rules in the filter match the detected packet, then the packet is, by default, ignored.

I have several RealSecure Engines that I need to set up and run with a particular filter and signature configuration. Is there a way I can configure once and transfer that configuration to other machines?

Yes, you can use the Template Configuration dialog to save your configuration and apply it to any Engine that is associated with your Console.

I've corrected the Console's and the remote machine's Engine Authentication files, (*.auth files), but I can't start the Engine. What's going on?

Upon changing the contents of the *.auth files, it is advisable to stop the Engine and Console to reload the authentication files.

Will RealSecure handle my T3 access to the Internet?

Yes, depending on the type of network to which the T3 is attached Currently, RealSecure operates only on Ethernet (10Mbps) networks, so if the T3 is connected to one or more Ethernet networks, you can use RealSecure on these Ethernet segments.

Appendix C: Database Schema

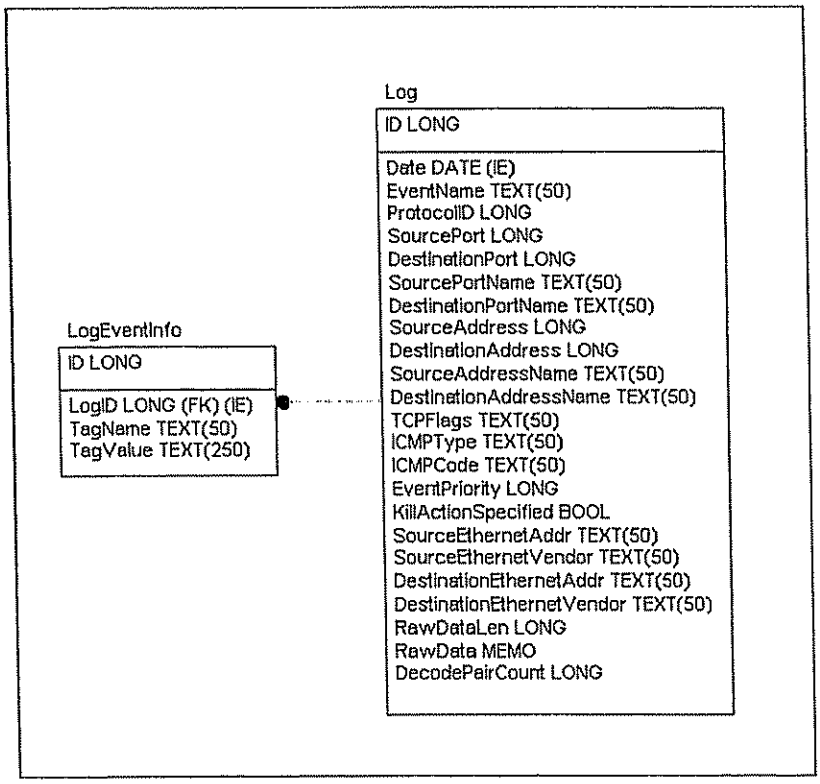


Figure 37: Database Schema

The database schema is provided to assist you in understanding the type of information collected by RealSecure's Database logs. The following section describes the various table entries.

Interpreting the Database Schema

Database information collected by RealSecure can be viewed through Microsoft Access.

CAUTION!

Do NOT modify the database schema or records. Such modifications can cause RealSecure's database to function improperly.

The following columns are present in the Log table:

ID: The unique identifier for the database record.

Date: The date and time the event was recorded.

EventName: The name of the recorded event as it appears in the Filters or Decodes/Attack Signatures.

ProtocolID: The protocol associated with the event. 0 represents TCP, 1 represents UDP, 2 represents ICMP, and 3 represents an unknown protocol.

SourcePort: The port number of the source.

DestinationPort: The port number of the destination.

SourcePortName: The name of the port of the source.

DestinationPortName: The name of the port of the destination.

SourceAddress: The IP address of the source.

DestinationAddress: The IP address of the destination.

SourceAddressName: The machine name of the source.

DestinationAddressName: The machine name of the destination.

TCPFlags: This column is not currently in use.

ICMPType: The type of ICMP packet.

ICMPCode: The code field from the ICMP packet.

EventPriority: The priority given to the event. 1 represents High Priority, 2 represents Medium Priority, and 3 represents Low Priority.

KillActionSpecified: Whether or not the engine is configured to kill a connection for an event of this type.

SourceEthernetAddr: This column is not currently in use.

SourceEthernetVendor: This column is not currently in use.

DestinationEthernetAddr: This column is not currently in use.

DestinationEthernetVendor: This column is not currently in use.

RawDataLen: The length of the rawData field value. If this value is not zero, it also indicates that the entry can be played back in the Session Playback window.

RawData: Raw data which is saved for later viewing through Session Playback.

DecodePairCount: The number of decode pairs written as log info records.

The following columns are present in the LogEventInfo table:

ID: The unique identifier of the entry in the Log table.

LogID: The ID of the entry in the Log table.

TagName: The name of the decode value.

TagValue: The actual value of the decode.

Index